REMARKS

The application has been reviewed in light of the Office Action dated May 19, 2004. Claims 9-39 are pending in this application with claims 9, 17, 22, 27, 28 and 29, 31 and 39 in independent form. New claims 31-39 correspond to previous claims 1-8 and 30, respectively, which were allowed in the Office Action dated August 14, 2001. The allowance of these claims was reiterated in the Office Action dated January 17, 2003. It is submitted that no new matter has been added and no new issues have been raised by the present Amendment.

Claim 9 was rejected under 35 U.S.C. § 103 (a) as allegedly unpatentable over U.S. Patent No. 5,493,682 to Morgenstern in light of U.S. Patent No. 5,493,682 to Tyra et al. Claims 10-16 were rejected under 35 U.S.C. § 103 (a) as allegedly unpatentable over Morgenstern in view of Tyra et al and further in view of U.S. Patent No. 5,325,531 to McKeeman et al. Claims 17, 22, and 27-29 were rejected under 35 U.S.C. § 103 (a) as allegedly unpatentable over Moregenstern in view of Tyra et al and further in view of U.S. Patent No. 5,926,819 to Doo et al. Claims 18-21 and 23-26 were rejected were rejected under 35 U.S.C. § 103 (a) as allegedly unpatentable over Morgenstern in view of Tyra et al. and further in view of Doo et al. and in view of Mckeeman et al.

Morgenstern, as understood by Applicants, relates to a method for processing heterogeneous data. More particularly, Morgenstern is relates to a method for heterogeneous data which uses an inoperability assistant module with specifications for transforming the data into a common intermediate representation of the data using the specifications and creating an information bridge with the interoperability assistant module through a process of program generation. See Morgenstern, column 1, lines 10-17.

Tyra et al., as understood by applicant relates to an apparatus and method for maintaining

computer software objects, and in particular is directed to a system based on object-oriented concepts using objects to represent system components and processes to be applied to those components.

Doo et al, as understood by Applicants, relates to database systems and more particularly to techniques for automatically performing a series of actions when an operation is applied to a body of data.

McKeeman et al., as understood by Applicants relates to computer aided software development. McKeeman et al. discloses a system including programs to implement, edit, compile, link and run sequences at very high speed. The compiler works on an incremental basis, line by line so if only one line is changed in an edit session, only that line need be recompiled.

Independent claim 9 of the present application relates to a method of generating a basic dependency tree of a code-object that does not take into consideration dependencies on triggers and on implementations of object oriented code objects. The method includes querying a database catalog for direct dependencies of a code object and then for each dependency found, doing the query recursively until all basic dependencies are generated into a dependency tree.

Independent claim 17 relates to a method of generating dependency information including dependencies of code objects on database triggers. The method includes using a recursive algorithm for querying a database catalog for direct dependencies of a code object and then for each dependency found, doing the query recursively until all basic dependencies are generated into a dependency graph; using a parser on each of the code objects in the dependency graph to identify DML statements that "fire" triggers so as to identify dependencies on triggers; applying the recursive algorithm on each of the triggers to incorporate the dependencies of the triggers into the dependency graph and repeating the previous steps for incorporating dependencies on triggers

and their dependencies until new dependencies are not added to the dependency graph.

The Office Action contends that Morgenstern discloses "querying a data catalog for direct dependencies of a code object" as recited in claims 9 and 17 of the present application. The Office Action cites Figure 4, column 20, lines 46-67 and column 21, lines 28-67 in support of this contention. The Office Action further states that a "database schema" is a kind of "database catalog." Further, in response to Applicants previous arguments, the Office Action contends that Morgenstern teaches development tools from CAD being applied to database schema on relational and object oriented databases and that a schematic structure graph from schema analyzers creates logical structure diagrams representing the dependency information of database. See Office Action, page 2. The Office Action further contends that the LCD is a uniform representation and represents schematic structure of data and is also a cycle directed graph and that it would arise when a data structure refers to itself recursively in its schema or data definition of a database. The Office Action further contends that it is a logical graph with logical dependencies. See id. Applicants respectfully disagree.

Morgenstern discloses a dependency graph that is the basis for rule execution and is implemented as a high level abstract class for the information bridge. The dependency graph is composed of three submodules for the input tree, rule graph and output tree. See Morgenstern, column 20, lines 47-50. The input tree includes a schema tree and an instance tree. See id. at lines 61-62. Morgenstern further discloses the integration of a wide variety of information resources including relational and object databases, CAD design tools, simulation packages, data analysis and visualization tools and other software modules while eliminating the need for specialized translators. Morgenstern discloses a declarative specification language that is utilized to represent source and target data representations. This high level data structure specification (HLDSS) provides a uniform language for representing diverse databases and file formats. See

Morgenstern, Column 3, line 7-20. Morgenstern further discloses a logical structure diagram (LSD) is a uniform representation of the HLDSS specification and represents the schematic structure of the data. See Morgenstern, column 13, lines 18-20. Morgenstern further discloses that the HLDSS is parsed and processed to create LSDs. See Morgenstern, column 11, lines 25-28.

Morgenstern, however, fails to show or suggest "querying a data catalog for direct dependencies of a code object" as recited in independent claims 9 and 17 of the present application.

Further, Applicants respectfully submit that Tyra et al. fails to show or suggest "querying a data catalog for direct dependencies of a code object".

Accordingly, it is respectfully submitted that independent claims 9 and 17, and the claims depending therefrom, are patentable over the cited art for at least the reasons mentioned above.

Claim 22 of the present application relates to a method of generating dependencies of code objects as well as implementations of object oriented code objects in a database. The method includes applying a recursive algorithm that queries a database for dependency information and outputs a direct dependency graph of a database code object, the "direct dependency graph" containing dependencies that do not involve dependencies on triggers and on implementations of object oriented code objects in the database; applying the recursive algorithm on each of the object oriented code objects in the dependency graph to incorporate dependencies of implementations of code objects in the database; and repeating the above steps for incorporating dependencies of implementation of object code objects until new dependencies are not added to the dependency graph.

The Office Action contends that Morgenstern discloses "applying a recursive algorithm that queries a database for dependency information" as recited in claim 22 and further contends

that a database schema is a kind of database catalog. Applicants respectfully disagree.

Morgenstern discloses building an input schema and output schema by analyzing respective high level data structure specifications HLDSS. Morgenstern further discloses that the information used to generate the chart of Fig. 4, which was cited by the Office Action, is defined by the HLDSS. See Morgenstern, column 20, lines 62-65.

It is respectfully submitted that Morgenstern fails to show or suggest "applying a recursive algorithm that queries a database for dependency information" as recited in claim 22 of the present application.

Accordingly, it is respectfully submitted that claim 22, and the claims depending therefrom, are patentable over the cited art for at least the reasons discussed above.

Claim 27 of the present application relates to a system for identifying dependencies; if any, of a target data base code object. The system includes a digital computer, a database service coupled to the computer; a data base coupled to the database server having data stored therein, the data including object oriented code objects, specifications of packages, implementations of packages, specifications of types, implementations of types and triggers; and a code mechanism for querying a database for dependency information and generating a dependency graph, the dependency graph being a data structure.

As noted above with respect to claim 22, it is respectfully submitted that Morgenstern fails to teach or suggest "applying a recursive algorithm that queries a database for dependency information." It is respectfully submitted that Morgenstern further fails to teach or suggest "a code mechanism for querying a database for dependency information" as recited in claim 27 of the present application.

Accordingly, it is respectfully submitted that claim 27 is patentable over the cited art for at least the reasons discussed above.

Claim 28 relates to a method for generating dependencies of a target data base code object, using a computer having a processor, memory, display, input/output devices, the method of including: providing a database coupled to the computer having data stored therein, the data including representations of object oriented code objects, specifications of packages, implementations of packages, specifications of types, implementations of types and triggers and using a recursive code mechanism for querying a database for dependency information and generating a dependency graph, the dependency graph being a data structure and having entries to contain representations of dependent code objects, specifications of packages, implementations of packages, specifications of types, implementations of types, triggers and dependencies of triggers which are relevant to the target data base code object.

As noted above with respect to claim 22, it is respectively submitted that Morgenstern fails to teach or suggest "applying a recursive algorithm that queries a database for dependency information." Similarly, it is respectfully submitted that Morgenstern fails to teach or suggest "using a recursive code mechanism for querying a database for dependency information" as recited in claim 28 of the present application.

Accordingly it is respectfully submitted that claim 28 is patentable over the cited art for at least the reasons noted above.

Claim 29 relates to a computer program product embedded on a computer readable medium for use in debugging a target data base code object including a recursive code mechanism for querying a database for dependency information and generating a dependency graph of the target data base code object, the dependency graph being a data structure and having entries to contain representations of depending code objects, specifications of packages, implementations of packages, triggers and dependencies of triggers which are relevant to the target data base code object; and a program code mechanism for using the dependency graph to

debug the target.

As previously noted with regard to claim 22, it is respectfully submitted that Morgenstern fails to teach or suggest "applying a recursive algorithm that queries a database for dependency information." It is respectfully submitted that Morgenstern fails to teach or suggest "a recursive code mechanism for querying a database for dependency information" as recited in claim 29 of the present application.

Accordingly it is respectfully submitted that claim 29 is patentable over the cited art for at least the reasons discussed above.

As noted above, new claims 31-39 correspond to previously allowed claims 1-8 and 30, respectively. Accordingly, it is respectfully submitted that new claims 31-39 are in condition for allowance.

The Office is hereby authorized to charge any additional fees that may be required in connection with this Amendment After Final Action to credit any overpayment to our Deposit Account No. 03-3125.

If a petition for an extension of time is required to make this response timely, this paper should be considered to be such a petition, and the Commissioner is authorized to charge the requisite fees to our Deposit Account No. 03-3125.

If a telephone interview could advance the prosecution of this application, the Examiner is respectfully requested to call the undersigned attorney.

Entry of this Amendment After Final Action and allowance of this application are respectfully requested.

Respectfully submitted,

RICHARD F. JAWORSKI
Reg. No. 33,515
KEITH J. BARKAUS
Reg. No. 51,431
Attorney for Applicants
Cooper & Dunham LLP
Tel.: (212) 278-0400